DISICO

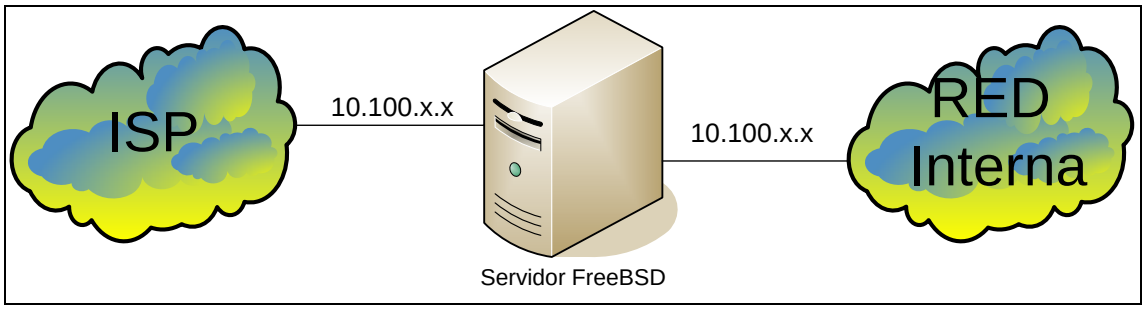
# Bridge + Monitoreo + Administración de Ancho de Banda

Manual

## Bridge + monitoreo + administración de ancho de banda

Uno de los mayores problemas con los cuales se encuentran los distintos administradores de red, es identificar que sucede con el consumo de ancho de banda al interior de esta, por otro lado son la aplicaciones particulares que tiene dependencia en la red o el hecho que cualquier implementación con un gran numero de usuarios debe ser transparente, de tal manera que no afecte la confianza.

La implementación física realizada se puede visualizar el la Figura.




Como se puede visualizar el segmento de red que ingresa desde el isp institucional no cambia al ingresar a la red interna como sucede cuando se usa NAT, en este caso es implementado nuestro servidor como un router tipo BRIDGE, esta implementación permite realizar un monitoreo o un firewal totalmente transparente al Usuario, para realizar este labor es necesario seguir los siguientes pasos:

### Configuración del Servidor

Después de haber realizado la instalación del sistema operativo FreeBSD para este tutorial se utilizó la versión 6.0, se deben configurar los parámetros básicos del computador

Solo se debe configurar una IP Válida en la interfaz externa o sea al borde donde se encuentre el ISP institucional, esta debe ser valida en el segmento de red, además se deben configurar todo los parámetros básicos como puerta de enlace y dirección del servidor de DNS

Una vez realizado esto instalamo por medio del paquete de ports actualizado, las siguientes aplicaciones que durante el desarrollo pueden llega a ser necesarias:

1. APACHE13 /usr/ports/www/apache13
2. RRDTOOL /usr/ports/databases/rrdtool
3. PHP4 /usr/ports/lang/php4

Con estas tres aplicaciones externas ya tenemos suficiente para la labor a realizar, el resto de las utilidades las obtendremos al momento de reconfigurar y recompilar el KERNEL de nuestro sistema Operativo.

Una vez terminada esta configuración es necesario recompilar nuestro kernel para esto es necesario seguir los siguientes pasos:

```
# cd /usr/src/sys/i386/conf/
# cp GENERIC MIKERNEL
```

Posterior a esto editamos MIKERNEL con el comando vi y agregamos las líneas como se describe a continuación

Editamos con:

```
# vi MIKERNEL
```

Y se agrega lo siguiente

```
options     BRIDGE                                  # linea alternativa
options     IPFIREWALL                              # Activa firewall ipfw
options     IPFIREWALL_VERBOSE
options     IPFIREWALL_VERBOSE_LIMIT
options     IPFIREWALL_DEFAULT_TO_ACCEPT
options     DUMMYNET    #permite realizar adminstracion de ancho de banda
```

Una vez ingresadas estas lineas sólo basta con guardar el archivo con los cambios esto depende del editos si se utilizó vi para esto se debe seguir la secuencia de

Esc :wq!

Con esto ya estamos listos para compilar el nuevo kernel, paras esto es necesario realizar los siguientes pasos:

```
# /usr/sbin/config MIKERNEL
# cd ../compile/MIKERNEL
# make depend
# make
# make install
```

Con esto ya se encuentra instalado el Nuevo Kernel del sistema operativo solo basta con reiniciar el servidor.

Con estos parámetros ya se cuenta con todo lo necesario para crear generar el sistema de monitoreo de ancho de banda de la red. Entonces se deben realizar las siguientes configuraciones

## 1. configurar el bridge

Para configurar el bridge es necesario editar el archivo /etc/sysctl.conf y agregar las siguientes líneas.

```
net.link.ether.bridge.enable=1
net.link.ether.bridge.config=em0:0,em1:0
net.link.ether.bridge.ipfw=1
```

Aquí se indica desde que interfaces se realizara el bridge siendo la primera de ellas la que cuenta con ip configurada, estas líneas pueden cambiar levemente dependiendo de la versión de freebsd que se este utilizando, debemos recordar que para este caso se esta utilizando la versión 6.0 o superior.

## 2. Reglas de conteo

Para poder contabilizar todo el trafico que pasa por nuestro servidor es necesario realizar algunas reglas de conteo con ipfirewall, esto lo podemos realizar el archivo /etc/rc.firewall donde se encuentra una estructura como a continuación

```
[Oo][Pp][Ee][Nn])
    setup_loopback
    ${fwcmd} add 65000 pass all from any to any
    ;;
```

Este es el caso de firewall tipo open, para nuestro caso sólo basta con realizar un nuevo caso siguiendo el mismo esquema, el realizado en el servidor es el siguiente:

```
[Ii][Gg][Oo][Rr])
    setup_loopback
    ${fwcmd} add 1000 count all from 10.100.6.166 to any
    ${fwcmd} add 1001 count all from any to 10.100.6.166
    ${fwcmd} add 1010 count all from 10.100.21.0/24 to any
    ${fwcmd} add 1011 count all from any to 10.100.21.0/24
    ${fwcmd} add 1020 count all from 10.100.26.0/24 to any
    ${fwcmd} add 1021 count all from any to 10.100.26.0/24
    ${fwcmd} add 1030 count all from 10.100.27.0/24 to any
    ${fwcmd} add 1031 count all from any to 10.100.27.0/24
    ${fwcmd} add 1040 count all from 10.100.28.0/24 to any
    ${fwcmd} add 1041 count all from any to 10.100.28.0/24
    ${fwcmd} add 1050 count all from 10.100.6.0/24 to any
    ${fwcmd} add 1051 count all from any to 10.100.6.0/24
    ${fwcmd} add 1060 count all from 10.100.34.0/24 to any
    ${fwcmd} add 1061 count all from any to 10.100.34.0/24
    ${fwcmd} add 1070 count all from 10.100.51.0/24 to any
    ${fwcmd} add 1071 count all from any to 10.100.51.0/24
    ${fwcmd} add 1080 count all from 10.100.61.0/24 to any
    ${fwcmd} add 1081 count all from any to 10.100.61.0/24
    ${fwcmd} add 1090 count all from 10.100.5.0/24 to any
    ${fwcmd} add 1091 count all from any to 10.100.5.0/24
    ${fwcmd} add 1092 count all from any to any in via em0
    ${fwcmd} add 1093 count all from any to any out via em0
```

```
${fwcmd} add 1094 count all from any to any in via em1
${fwcmd} add 1095 count all from any to any out via em1
${fwcmd} add 55100 pipe 1 all from any to any in via em0
${fwcmd} add 55200 pipe 2 all from any to any in via em1
${fwcmd} pipe 1 config bw 10Mbit/s queue 10Kbit
${fwcmd} pipe 2 config bw 10Mbit/s queue 10Kbit
#${fwcmd} add 65000 pass all from any to any
;;
```

Como se puede apreciar casa una de las reglas permite contar todos los paquetes que son transmitidos y decepcionados para cada segmento de red esto lo encontramos en las reglas que ocurren entre la 1000 y la 1091, estos números son a gusto del creador, lo que si no se deben repetir el orden afecta directamente al funcionamiento de las reglas. Pero esto es tema de firewall para nuestro no.

Las reglas 1092,1093,1094,1095, permiten contar el trafico total que ocurre en toda la red, pero de estas quien almacena con exactitud el conteo de los paquetes entrantes y salientes desde nuestra red son la regla 1092 y la 1094 , las otras dos reglas cometen errores de conteo debido a que se encuentran en el cambio de interfase, ahora una vez ya creadas nuestra reglas es necesario activar el firewall, de tal manera que cada vez que encienda se generen de forma automática cada una de estas, para esto es necesario agregar las siguientes líneas al archivo /etc/rc.conf

```
firewall_enable="YES"
firewall_type="IGOR"
firewall_quiet="YES"
```

Con esto ya esta too listo, ahora reiniciar la maquina y esperar unos momentos a que vuelva a encender recordar que para esto solo basta escribir reboot y presionar enter.

Al encender nuevamente la maquina ingresamos con el login y pasword respectivos de root y ejecutamos el siguiente comando

```
igor# ipfw show
```

Con este comando se obtiene la siguiente salida

```
01000     54587     8862211 count ip from 10.100.6.166 to any
01001     29791    43469116 count ip from any to 10.100.6.166
01010         0           0 count ip from 10.100.21.0/24 to any
01011         0           0 count ip from any to 10.100.21.0/24
01020  148620105 122566353587 count ip from 10.100.26.0/24 to any
01021  130014661  69429760766 count ip from any to 10.100.26.0/24
01030   86889300  61971288947 count ip from 10.100.27.0/24 to any
01031   90111701  62848531402 count ip from any to 10.100.27.0/24
01040 1310675630  78944852678 count ip from 10.100.28.0/24 to any
01041  109149222  94806175626 count ip from any to 10.100.28.0/24
01050  192328617  73695008488 count ip from 10.100.6.0/24 to any
01051  211919331 139081693241 count ip from any to 10.100.6.0/24
01060         0           0 count ip from 10.100.34.0/24 to any
01061       518      106549 count ip from any to 10.100.34.0/24
01070    933181    214828050 count ip from 10.100.51.0/24 to any
01071   1178787   1046642090 count ip from any to 10.100.51.0/24
```

```
01080        0          0 count ip from 10.100.61.0/24 to any
01081      960     101348 count ip from any to 10.100.61.0/24
01090 19028950 5798932790 count ip from 10.100.5.0/24 to any
01091 20719240 14038694913 count ip from any to 10.100.5.0/24
01092 540746462 376454601848 count ip from any to any in via em0
01093    98482    82928674 count ip from any to any out via em0
01094 1752874203 340801800725 count ip from any to any in via em1
01095        3         208 count ip from any to any out via em1
55100 540746462 376454601848 pipe 1 ip from any to any in via em0
55200 1752874203 340801800725 pipe 2 ip from any to any in via em1
65535    103273    83316728 allow ip from any to any
```

Si se observa bien, cada una de estas lineas son la que se han configurado en el archivo /etc/rc.firewall, lo que indica que todo esta correcto hasta el momento, cada una de las lineas cuenta con 5 columnas de la forma:

| Columna1 | Columna2 | Columan3 | Columna4 | Columna 5 |
|---|---|---|---|---|
| 01050 | 192328617 | 73695008488 | count ip from | 10.100.6.0/24 to any |

Donde

- Columna1 numero de la regla
- Columna 2 cantidad de paquetes
- Columna 3 byte transmitidos
- Columna 4 regla
- Columa 5 segmento de red a evaluar

Con estos datos ya se tiene suficiente información para poder realizar el monitoreo del consumo de ancho de banda de los distintos segmentos de red.

## 3. Administración del ancho de banda

Uno de los mayores problemas que ocurren en toda red es el abuso del consumo de ancho de banda de parte de los usuarios, debido a u sinumero de aplicaciones del tipo p2p que dentro de sus mayores ventajas como aplicación esta la utilización de la totalidad del ancho de banda disponible de la red lo que perjudica al resto de los usuarios, para esto se utiliza la capacidad de DUMMYNET que fue configurado en el KERNEL recompilado, este se utiliza como una propiedad de IPFIREWALL permitiendo con esto crea pipas que no permitan saturar al ISP, su utilización es muy simple y se puede apreciar en las reglas 55100 y 55200

```
${fwcmd} add 55100 pipe 1 all from any to any in via em0
${fwcmd} add 55200 pipe 2 all from any to any in via em1
${fwcmd} pipe 1 config bw 10Mbit/s queue 10Kbit
${fwcmd} pipe 2 config bw 10Mbit/s queue 10Kbit
```

Donde se crea un pipe para todo lo que ingrese vía la interfaz em0 y otra para todo lo que salga desde nuestra red o sea todo lo que ingrese a la interfaz em1, posteriormente solo basta con indicar a cada una de las distintas pipas realizadas cual es el máximo ancho de banda que puede transmitir, con esto se permite que en caso de un abuso de consumo este queda totalmente aislado a nuestra red interna y se pueda detectar el problema por medio del monitoreo.

## 4. Monitoreo

Para realizar el monitoreo es necesario primero contar con un lugar donde se encuentren nuestros datos para esto se ha utilizar RRDTOOL o round robin database tool, que no es nada mas que un motor de bades de datos que cada cierto periodo elimina los datos, manteniendo sólo el historial solicitado, son mucho los tipos de aplicaciones que pueden ser utilizados, debido a que su configuración y la creación de las tablas esta basada en un espectro de periodicidad.

Para facilitar la creación de las distintas tablas se puede utilizar el siguiente script

```
rrdtool create todos.rrd -s300
DS:intodos:COUNTER:600:0:U
DS:outtodos:COUNTER:600:0:U
RRA:AVERAGE:0.5:1:576
RRA:AVERAGE:0.5:12:1440
RRA:AVERAGE:0.5:288:730
```

Se crea la base de datos todos.rrd que almacena datos cada 300 segundos (-s300),
Se describen las columanas por medio del comando
DS:nombrecolumna:TIPO:TIEMPO:MIN:MAX

Para este caso se indica que el nombre de la columna es intodos que es de tipo COUNTER y que pueden pasar hasta 600 segundos antes de declarar el dato como desconocido, el valor mínimo que puede ser ingresado es 0 y el máximo es U o infinito

No daremos mas detalles de la creación de la tabla para no complicar este tutorial, ahora para el proceso de almacenamiento se deben indicar por cuanto tiempo estarán

almacenados los valores y qye tipo sea almacenado, en este caso se guarda un promedio o AVERAGE de siguiente forma
RRA:AVERAGE:0.5:1:576

Esto indica serán almacenados los promedios directos durante 48 horas o sea para explicar mejor el valor 0.5 es un factor de recuperación de los datos, no muy bien explicado por sus autores, hasta el momento de la escritura de este tutorial, el 1 indica que cada 30 seg será almacenado un dato esto ocurrirá por 576 intervalos o sea 30 seg * 576 lo que entrega 48 horas.

En el caso dos indica que se almacena un promedio cada 1 hora y esto es almacenado durante 1440 intervalos lo que nos indica que estos promedios son almacenados durante 2 meses

El caso tres es análogo el promedio de dos meses de transacciones es almacenado durante 2 años con esto se evita saturar las distantas bases de datos y mantener un promedio de las transacciones

Ahora para poder visualizar los distintos tráficos es necesario primero indicar al sistema que almacene los datos extraídos desde el IPFW, para poder extraer todos los datos se utiliza el siguiente script

```
#!/usr/local/bin/bash
basename=`basename $0`
rrdname='/root/traficomio.rrd'
template=''
values='N'
debug=''
addValue(){
      if [ $debug ]; then  echo "Adding $name line $line"; fi
      if [ -z $template ]; then template=$name;  else template="$template:$nam
e"; fi
      counter=`echo "$line" | awk '{ print \$3; }'`
      values="$values:$counter";
      echo $values;

      }
IFS="
"
for line in `/sbin/ipfw show`; do
      name=''
      case $line in
            01000* ) name='inmio';;
            01001* ) name='outmio';;

      esac
      echo $line;
      echo $name;
      if [ -n "$name" ]; then addValue; fi

done
```

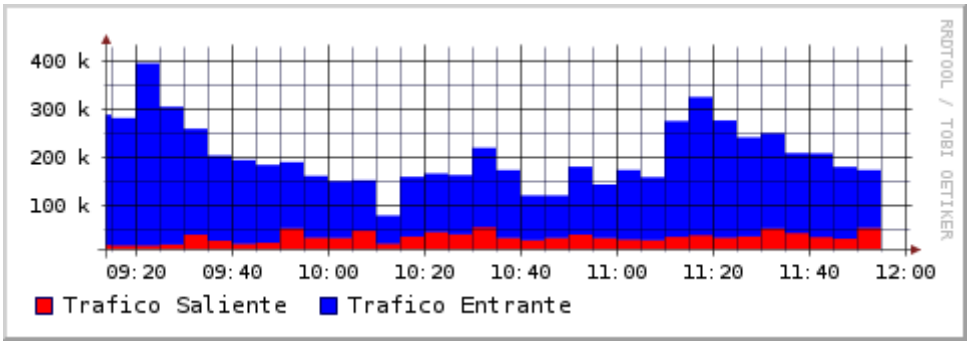
```
if [ $debug ]; then
        echo "Template=$template"
        echo "Values=$values"
fi
echo $values;
if [ -n "$template" ]; then /usr/local/bin/rrdtool update "$rrdname" -t "$templa
te"  "$values"
else echo "$basename: No known lines found." >&2; fi
```

Este strip permite almacenar los datos de la regla respectiva a la base de datos traficomio.rrd, ahora es necesario contar con otro script que permite al sistema generar los distintos gráficos del trafico de la red, para esto se utiliza el siguiente script.

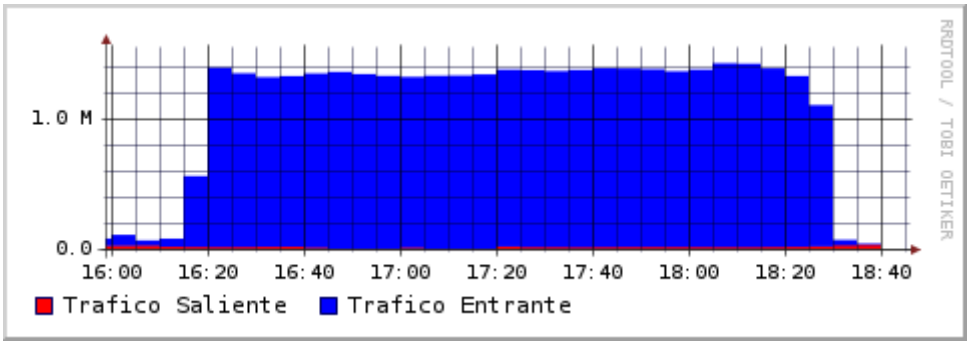
```
#!/usr/local/bin/bash

`/usr/local/bin/rrdtool graph /usr/local/www/data/graficos/mes/unmes280.png -a P
NG -s -2592000 -c BACK#ffffff -c CANVAS#ffffff -c GRID#000033 -c MGRID#000000 -c
 FONT#000000 -c FRAME#000066 DEF:in280=/root/10100280.rrd:in280:AVERAGE DEF:out2
80=/root/10100280.rrd:out280:AVERAGE AREA:in280#ff0000:"Trafico Saliente" STACK:
out280#0000ff:"Trafico Entrante"`;
```

Es solo una línea de comando en la que deben ser reemplazados los valores deseados con esta obtenemos gráficos del siguiente tipo




O detectar sobre consumo por aplicación p2p



Todos esto puede se implementado en una pagina web con apache en el servidor y permite que la aplicación pueda ser visualizada desde cualquier navegador en forma dinámica.